

Uwe Krause

(wip) Noch kein Titel

Fakultät Informatik und Digitale Gesellschaft

Inhaltsverzeichnis

1 Motivation & Vorhaben	1
2 Didaktische Basis	2
2.1 Blooms Taxonomie	2
2.2 ICAP (Interactive - Constructive - Active - Passive)	3
2.3 Methoden der Wissensvermittlung	4
2.4 Interaktion mit Sicherheitslücken (Serious Games)	5
3 Interaktive Lernumgebung	6
3.1 Lernziele	6
3.2 Zu vermittelnde Sicherheitslücke: Broken Access Control	7
3.3 Szenario	7
3.3.1 Glesn	7
3.3.2 Aufbau der Anwendung	8
3.4 Sicherheitslücken in der Anwendung	9
3.4.1 Auslesen von Meta-Daten trotz fehlender Zugriffsberechtigung	10
3.4.2 Rechteauserweiterung	10
3.5 Ausnutzen der Sicherheitslücken	11
3.6 Beta-Test	13
3.7 Durchführung eines Workshops	13
4 Fazit & Ausblick	14
4.1 Fazit	14
4.2 Ausblick	14
Bibliografie	15

1 Motivation & Vorhaben

Erfolgreiche Angriffe auf IT-Systeme verursachen durch Geschäftsunterbrechung und Wiederherstellungszeiten, Reputationsverlust und eventuelle Bußgelder enorme Kosten. Aktuell wird von einer durchschnittlichen Schadenshöhe von 8,79 Millionen¹ US Dollar für einen auf eine Sicherheitslücke zurückführbaren Datenschutzverstoß ausgegangen (IBM DEUTSCHLAND, 2024). Da die Ursachen für Sicherheitslücken in IT-Systemen unsichere Programmierung oder Programmierfehler sind, kann die Verbesserung der Qualität von Programmcode die Eintrittswahrscheinlichkeit eines Datenschutzverstoßes reduzieren.

Eine Möglichkeit der Qualitätsverbesserung – direkt bei den Menschen ansetzend, die täglich potentiell problematischen Code generieren – ist das Schaffen oder Erhöhen eines Problembewusstseins, welche direkten Auswirkungen es haben kann, wenn durch unvorsichtige Programmierung unsicherer Code erzeugt wird. Erleben Programmierer:innen selbst, wie leicht Sicherheitslücken im Programmcode ausnutzbar sind, ist zu erhoffen, dass sie mehr Verständnis für sichere Programmiermethoden aufbauen. Basierend auf der Annahme, dass Lernende am erfolgreichsten sind, wenn sie sich aktiv mit dem Lerngegenstand beschäftigen, ist nach einer geeigneten Methode zu suchen, mit der Programmierer:innen sich möglichst intensiv mit Sicherheitslücken beschäftigen.

Durch Anwendung der ICAP-Methode (Interactive, Constructive, Active, Passive) (CHI & WYLIE, 2014) soll erreicht werden, dass Entwickler:innen ein gemäß Blooms (überarbeiteter) Taxonomy (ANDERSON & KRATHWOHL, 2001) vertieftes Verständnis nicht nur einer einzelnen Sicherheitslücke, sondern einer ganzen Kategorie an Sicherheitsproblemen erhalten. Dies wiederum soll dafür sorgen, dass sie künftig die Kompetenzen erhalten, diese und ähnliche Unachtsamkeiten nicht nur nicht mehr in eigenem Code zu produzieren, sondern auch in fremden Code identifizieren zu können.

Das Ziel soll zwar nicht die Weiterbildung von Entwickler:innen zu Penetrationstester:innen sein, jedoch lassen sich Elemente aus verbreiteten Trainingsmethoden von IT-Sicherheits-Expert:innen für den Lernprozess der Qualitätsverbesserung entleihen. Als eines der in der Aus- und Weiterbildung von Penetrationstester:innen verwendeten Vorgehen können Übungen nach dem Vorbild von *capture the flag*² verwendet werden. Die Anwendung dieser *serious games* (COSTA & RIBAUDO, 2023) zeichnet sich durch einen hohen Interaktionsgrad aus, weshalb davon auszugehen ist, dass investierte Zeit

¹4,46 Mio. USD „Unbekannte Zero-Day-Sicherheitslücke“ plus 4,33 Mio. USD „Bekanntermaßen ungepatchte Sicherheitslücke“ (IBM DEUTSCHLAND, 2024)

²Siehe beispielsweise (EAGLE & CLARK, 2004), (MCDANIEL u. a., 2016), (TSEPELIS, 2025)

effizient genutzt wird, um das IT-Sicherheits-Problembewusstsein von Entwickler:innen zu schärfen.

Die weitverbreitetste Kategorie von Sicherheitslücke bei der Programmierung von Internetseiten ist eine mangelhafte Zugriffskontrolle (OWASP, 2025). Damit die Auswirkungen von Programmierfehlern dieser Art besser greifbar werden, wird in dieser Arbeit eine Internetseite entwickelt, die absichtlich Fehler aus dieser Kategorie enthält. Diese *Challenge* wird in einem Workshop verwendet, um Teilnehmer:innen dieses Problemfeld näherzubringen.

2 Didaktische Basis

Die gewünschte Erhöhung des Problembewusstseins entspricht letztlich einem Dazulernen der Programmierer:innen. Verschiedene didaktische Konzepte bekräftigen, dass Lernende am erfolgreichsten sind, wenn sie sich aktiv mit dem Lerngegenstand beschäftigen.

2.1 Blooms Taxonomie

Gemäß Blooms (überarbeiteter) Taxonomie (ANDERSON & KRATHWOHL, 2001) werden verschiedene aufeinander aufbauende Ebenen des Verständnisses identifiziert. Diese werden oft wie in Abbildung 1 als Pyramide dargestellt. Die Basis bildet das „Erinnern“, die Fähigkeit relevantes Wissen aus dem Langzeitgedächtnis abzurufen. Darauf aufbauend folgt das „Verstehen“, das es ermöglicht Wissen zusammenzufassen, zu interpretieren, zu vergleichen und zu erklären. Die Stufe des „Anwendens“ enthält das selbstständige Ausführen von Gelerntem. Darauf folgt die Fähigkeit der „Analyse“ von Sachverhalten, also die Zerlegung des Materials in seine einzelnen Komponenten, Identifikation übergreifender Struktur und Zweck sowie das Erkennen von Mustern. Erst wer dazu in der Lage ist, wird überhaupt die Grundlagen haben, um fachlich „bewerten“ zu können, sich ein Urteil zu bilden, bis hin dazu, Bestehendes zu kritisieren. An der Spitze der Taxonomie steht die Fähigkeit angeeignete Elemente zusammenzufügen, um Neues zu erschaffen.

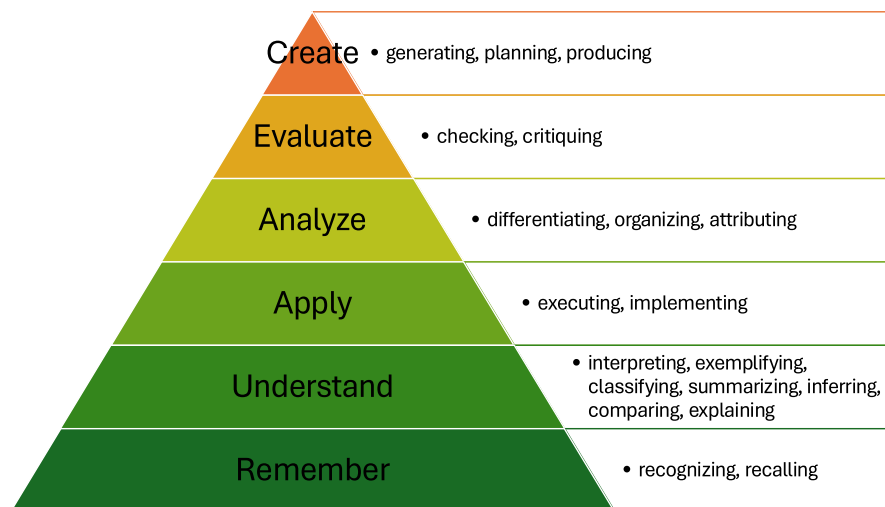


Abbildung 1: Blooms Pyramide, nach (ANDERSON & KRATHWOHL, 2001). Eigene Darstellung

2.2 ICAP (Interactive - Constructive - Active - Passive)

Bei dem *ICAP*-Ansatz (CHI & WYLIE, 2014) werden aufbauend auf dem Vorgehen des *aktiven Lernens* vier Modelle kognitiver Lernaktivitäten definiert und Wege vorgeschlagen, wie Wissensveränderungsprozesse angestoßen werden sollten: Zwar kann Wissen durch passives Empfangen von Informationen beim Zuhören oder Lesen aufgebaut werden, dies stellt jedoch die niedrigste Stufe dar. Die Wissensaufnahme kann aktiver gestaltet werden, zum Beispiel durch das Anfertigen zusätzlicher Notizen während des Zuhörens, dem Anstreichen von Textstellen während des Lesens oder durch Zusammenfassen einzelner Passagen. Konstruktives Verarbeiten erfolgt, wenn über die Aufnahme von Erlerntem hinausgehend selbst etwas generiert wird, beispielsweise wenn in einer Vorlesung Fragen gestellt werden oder zu einem beschriebenen Zusammenhang eine Zeichnung angefertigt wird. Die höchste Stufe des Modells wird erst durch Interaktion mit dem Lerngegenstand erreicht, beispielsweise durch Verteidigung eines Standpunkts in einer Gruppendiskussion, in deren Verlauf neue Ideen entstehen.

Diesen Lernaktivitäten werden die jeweilig zu erwartenden kognitiven Ergebnisse gegenübergestellt: Passiv erlangtes Wissen kann wörtlich und im gleichen Kontext wiedergegeben werden, wohingegen aktiv Erlangtes auf ähnliche Kontexte angewendet werden kann. Einmal generiertes Wissen kann in neuen Situationen angebracht werden, wobei Kenntnis über das erlernte Konzept Interpretation und Erklärungen erlaubt, auch

unter geänderten Rahmenbedingungen. Das tiefste Verständnis ist nach der Interaktion zu erwarten: Potentiell entstehen neue Produkte, Interpretationen und Ideen. (CHI & WYLIE, 2014)

2.3 Methoden der Wissensvermittlung

Die Essenz und ein in diversen weiteren Ansätzen³ immer wieder zu beobachtende Muster ist die möglichst aktive Beschäftigung mit dem Lerngegenstand durch die Lernenden selbst. Wenn Jura-Studierende Gerichtsurteile aufarbeiten, im Betriebswirtschaftsstudium Fallstudien bearbeitet werden und angehende Mediziner:innen anhand der Begutachtung von Leichen die Todesursache ermitteln (ECKSTEIN u. a., 2002), wie können Programmierer:innen sich interaktiv mit den abstrakten Themen der IT-Sicherheit und sicheren Programmierung beschäftigen?

Wird versucht Konzepte der IT-Sicherheit über Vorträge oder Webinare zu IT-Sicherheit zu vermitteln, stellen diese lediglich die niedrigste Stufe der ICAP-Skala dar. Mehr als eine grundlegende Wissensaufnahme ist nicht zu erwarten, was im Bloomschen Modell ebenfalls der untersten Stufe entspricht. Eine aktivere Beschäftigung mit sicheren Programmiermethoden kann etwa bei *code reviews* das Hervorheben von fehlerhaftem Code sein. Werden Problembereiche auf diese Art erkannt, entspricht dies dem „Verstehen“ auf der Bloomschen Skala. Es darf erwartet werden, dass das Wissen ausreichend abstrahiert wurde, sodass die Chance besteht, ähnliche Fehler in ähnlichen Kontexten erneut zu identifizieren. Um sich konstruktiv mit IT-Sicherheit zu beschäftigen, bieten sich *Hackathons* an, bei denen sicherheitsrelevante Systeme oder Komponenten entwickelt werden. Eine Person, die bereits einmal etwa ein Login-System entworfen oder implementiert hat, wird benötigte Verfahren nicht nur angewendet haben, sondern wahrscheinlich das Wissen auch in seine Bestandteile zerlegt und zumindest Teile davon analysiert haben. Sie sollte diese Fähigkeit auch unter geänderten Rahmenbedingungen anwenden können.

Dieser Logik folgend wird die Interaktion mit einer Sicherheitslücke selbst zu den drei Stufen des tiefsten Verständnis führen: Eine Person, die selbst in die Rolle einer Angreifer:in geschlüpft ist und es geschafft hat eine Sicherheitslücke auszunutzen, hat hierdurch bewiesen, die Lücke nicht nur analysiert, sondern auch umfassend bewertet zu

³Vgl. “Active Student”, “Students Decide”, “Honor Questions”, “Test Tube”, “Try It Yourself”, “Prefer Writing”, “Role Play”, “War Game”, “Invisible Teacher”, in (ECKSTEIN u. a., 2002).

haben. Wurde mit diesem Wissen sogar ein erfolgreicher Angriff erschaffen, so darf davon ausgegangen werden, dass das zugrundeliegende Problem wirklich verstanden wurde.

2.4 Interaktion mit Sicherheitslücken (Serious Games)

Der Umgang mit derartigen Sicherheitslücken sollte nicht mit Produktivsystemen ausprobiert werden. Die skizzierte Form der Interaktion kann aber als Thema sogenannter *serious games* behandelt werden. Allgemein werden diese entwickelt, um über reale Themen wie Gesundheitsfürsorge, Sicherheit oder Umweltfragen aufzuklären, zu schulen oder zu informieren (COOPER & BUCCHIARONE, 2023). Wenn serious games das Lernen komplexer und technischer Themen erleichtern, können Konzepte der IT-Sicherheit ebenfalls mit Techniken aus diesem Bereich vermittelt werden. In Lernumgebungen für konkrete zu lernende Konzepte der IT-Sicherheit stellen Spieler sich auf diese Weise realistischen Herausforderungen. (COSTA & RIBAUDO, 2023)

Werden auf diese Art Techniken der IT-Security vermittelt, findet dies oft im Rahmen von *capture the flag* statt, einer Art IT-Security-Wettbewerb (EAGLE & CLARK, 2004; MCDANIEL u. a., 2016). Hier gibt es verschiedene Typen von Aufgaben (*challenges*), beispielsweise im Stil der US-Amerikanischen Quizshow *Jeopardy*, bei der es für gelöste Aufgaben in unterschiedlichen Kategorien Punkte zu gewinnen gibt (TSEPELIS, 2025). Aufgaben in der Kategorie „Web“ bestehen üblicherweise aus einem Webserver, der den Spielern eine Website mit einer oder mehreren absichtlich platzierten Schwachstellen zur Verfügung stellt. Diesen Webserver gilt es anzugreifen, indem die Sicherheitslücke(n) identifiziert und ausgenutzt werden.

Es ist zu beobachten, dass Teilnehmende großes Interesse am wettbewerbsorientierten Lernen zeigen (SIAMI NAMIN u. a., 2016) und capture-the-flag-Aufgaben die Interaktion von Lernenden mit dem Lerngegenstand erhöhen, was wiederum zu besser entwickelten Fähigkeiten im Umgang mit Sicherheitslücken führt (LEUNE & PETRILLI, 2017).

Capture the Flag als Eigenart der serious games bieten Teilnehmenden eine Möglichkeit anhand extra hierfür bereitgestellter Systeme den praktischen Umgang mit Sicherheitslücken zu erlernen. Im Folgenden wird davon ausgegangen, dass dies also ein geeignetes Mittel ist, um Kenntnisse im Bereich der IT-Sicherheit zu vermitteln und nachhaltig zu festigen.

3 Interaktive Lernumgebung

Im praktischen Teil dieser Arbeit wird eine Lernumgebung dokumentiert, innerhalb derer die Interaktion mit einem weit verbreiteten und für Programmierer:innen direkt relevanten IT-Sicherheitsproblem möglich ist. Anhand des aufgebauten Systems soll ein serious game durchführbar sein, dessen Ziel darin besteht, eine bewusst platzierte Schwachstelle auszunutzen. Dies soll ein tiefes Verständnis für die Problematik der platzierten Sicherheitslücke fördern.

3.1 Lernziele

Auch wenn das Ziel dieses serious games das Ausnutzen einer Sicherheitslücke ist, so ist dies nicht das eigentliche Lernziel. Vielmehr sollen Teilnehmer:innen die in der Lernumgebung integrierten Sicherheitslücken zukünftig kennen (*remember*). Sie sollen verstanden haben, dass es Konsequenzen haben wird, wenn eine Angreifer:in diese Lücke ausnutzt (*understand*). Eigener Code sollte zukünftig frei von diesem Problem sein (*apply*). Teilnehmer:innen sollen zudem in der Lage sein, eigenen oder fremden Code daraufhin zu analysieren, ob er diese oder eine ähnliche Sicherheitslücke enthält (*analyze*). Sie sollten bewerten können, ob verwundbarer Code ausgenutzt werden kann (*evaluate*). In einer etwaigen Diskussion sollen sie letztlich ihr Gegenüber davon überzeugen können, dass diese Lücke geschlossen werden muss (*create*).

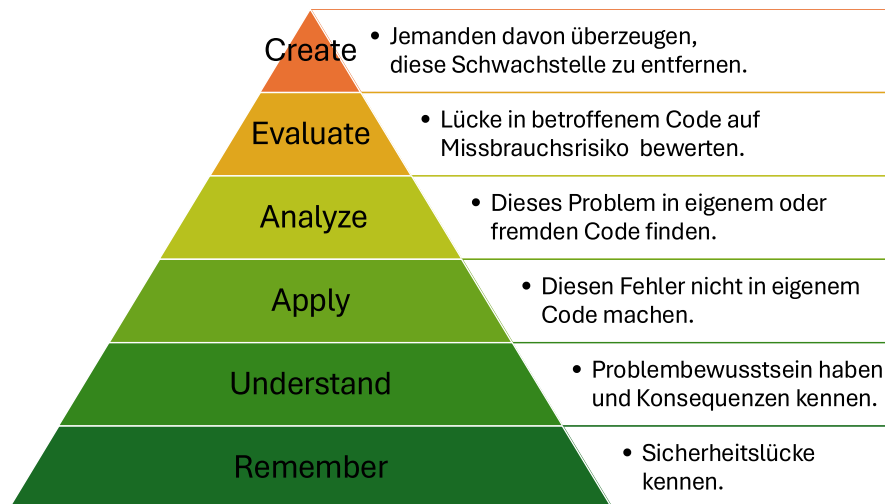


Abbildung 2: Lernziele in Relation zu Blooms Pyramide. Eigene Darstellung

3.2 Zu vermittelnde Sicherheitslücke: Broken Access Control

Als Versuch ein einheitliches Vokabular innerhalb von Organisationen und der IT-Industrie insgesamt zu schaffen, sammelt die not-for-profit Organisation MITRE⁴ Schwachstellen in einer „Common Weakness Enumeration“, CWE-Liste (MITRE, 2024). In den OWASP „Top 10 Web Application Security Risks“ werden Sicherheitslücken kategorisiert und diese Kategorien werden wiederum nach Inzidenzrate geordnet. Auf Platz 1 der schwerwiegendsten Sicherheitsrisiken sowohl im Betrachtungszeitraum 2017 – 2021 als auch 2021 – 2025 befinden sich Fehler der Kategorie „Mangelhafte Zugriffskontrolle“. Zugriffskontrollen bezeichnet Mechanismen, die sicherstellen, dass Nutzer:innen nur innerhalb ihrer vorgesehenen Berechtigungen handeln können. Werden diese Zugriffskontrollen mangelhaft implementiert, kann eine erfolgreiche Ausnutzung dazu führen, dass eine Angreifer:in Daten unbefugt lesen, ändern oder zerstören kann oder eine Anwendung dazu bringen kann, Dinge zu tun, die die Nutzer:in nicht können sollte. (OWASP, 2021; 2025)

3.3 Szenario

Die Entstehung der Umgebung soll in etwa der Entstehung einer realen Anwendung folgen. In der anzugreifenden Umgebung soll sich eine fiktive Anwendung befinden, die an und für sich korrekt funktioniert, aber unter Ausnutzung der Sicherheitslücken zu einem (aus Sicht einer fiktiven Organisation) fehlerhaften Verhalten gebracht werden kann.

Für dieses Szenario wird von einer Entwickler:in ausgegangen, die eine Kommunikationsplattform betreiben möchte. Angelehnt an die bestehenden Plattform „Reddit“, bei der sich Nutzer:innen zu vielen verschiedenen Themen öffentlich austauschen können, soll die neue Plattform einen themenbezogenen Austausch in vertraulichen Kleingruppen ermöglichen. Diese Kleingruppen sind im Gegensatz zu Reddit nicht öffentlich, sondern können nur auf Einladung eines Mitglieds betreten werden.

3.3.1 Glesn

Die Entwickler:in kreiert die Konkurrenzplattform „*Glesn*“⁵, welche Anwender:innen verspricht, sich innerhalb eines geschützten *Space* mit Anderen austauschen zu können.

⁴Finanziert unter anderem vom U.S. Department of Homeland Security.

⁵Bayrisch, vulg. für „gelesen“, Anspielung auf „reddit“, was selbst durch ein Wortspiel mit „I read it on reddit.“ entstanden ist (REDDIT.COM, 2006).

Innerhalb der Spaces können Anwender:innen *Artikel* schreiben, die von Anderen nur dann gelesen werden können, wenn diese selbst Teil des Space sind, in dem der Artikel veröffentlicht wurde.

3.3.2 Aufbau der Anwendung

Das User-Interface der Website besteht aus einem Kopfbereich und einem Bereich für den anzuzeigenden Inhalt. Der Kopfbereich ist aufgeteilt in einen Unter-Bereich für die Account-Verwaltung und einen für die Navigation.

Anwender:innen können auf der Seite einen Account anlegen, in dem die Berechtigungen (Zugehörigkeit zu Spaces) hinterlegt sind.⁶

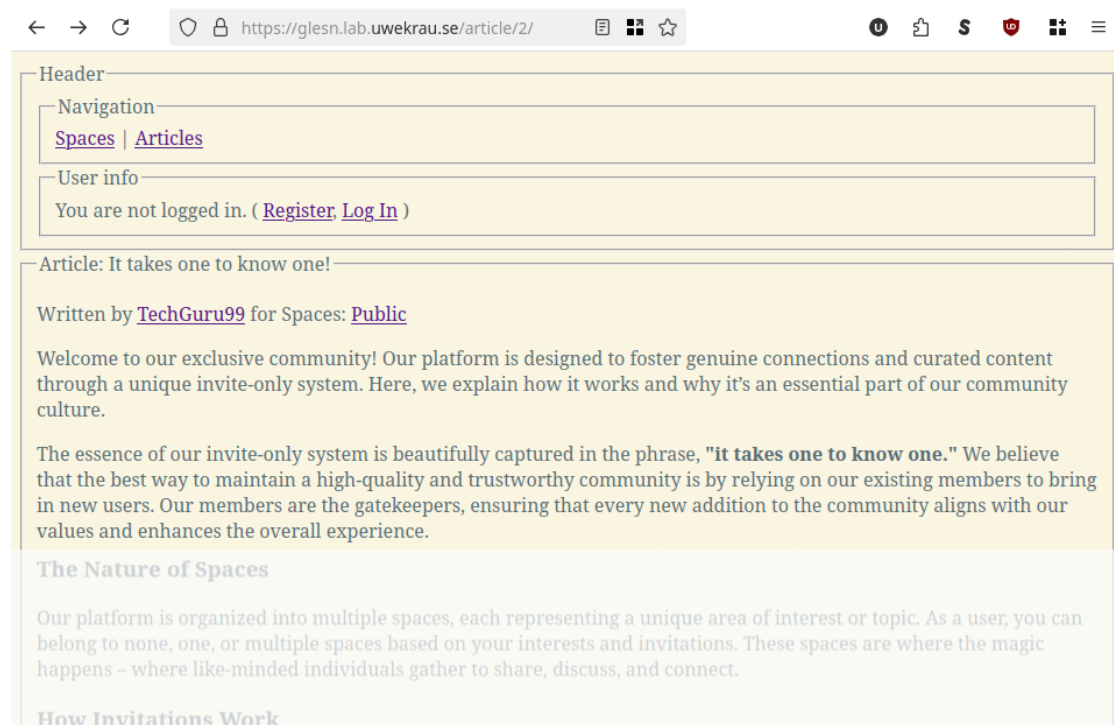


Abbildung 3: Screenshot der Website. In einem öffentlich einsehbareren Artikel wird das Funktionsprinzip erklärt.

Eine Einstiegsseite erklärt das Prinzip der Seite, also dass Anwender:innen innerhalb geschützter Spaces Artikel anlegen können und dass andere Anwender:innen diese Artikel

⁶Stellvertretend für Systemen mit einer Rollen-Basierten Zugriffskontrolle (*RBAC*, “Role-Based Access Control”).

nur lesen können, wenn sie zuvor von einer Person innerhalb des Space zu diesem eingeladen wurden.

Über die Navigation kann jeweils eine Übersicht aller für den verwendeten Account sichtbaren Spaces und Artikel aufgerufen werden.

Die einzelnen Artikel bestehen aus einer Überschrift, der Information, welche Autor:in den Artikel verfasst hat, welchen Spaces der Artikel zugewiesen ist und dem eigentlichem Inhalt des Artikels.

Möchte man mehr über die einzelnen Autor:innen erfahren, kann eine Übersicht ihrer Artikel aufgerufen werden. Für angemeldete Anwender:innen besteht außerdem die Möglichkeit, die ausgewählte andere Person zu einem der existierenden Spaces einzuladen.

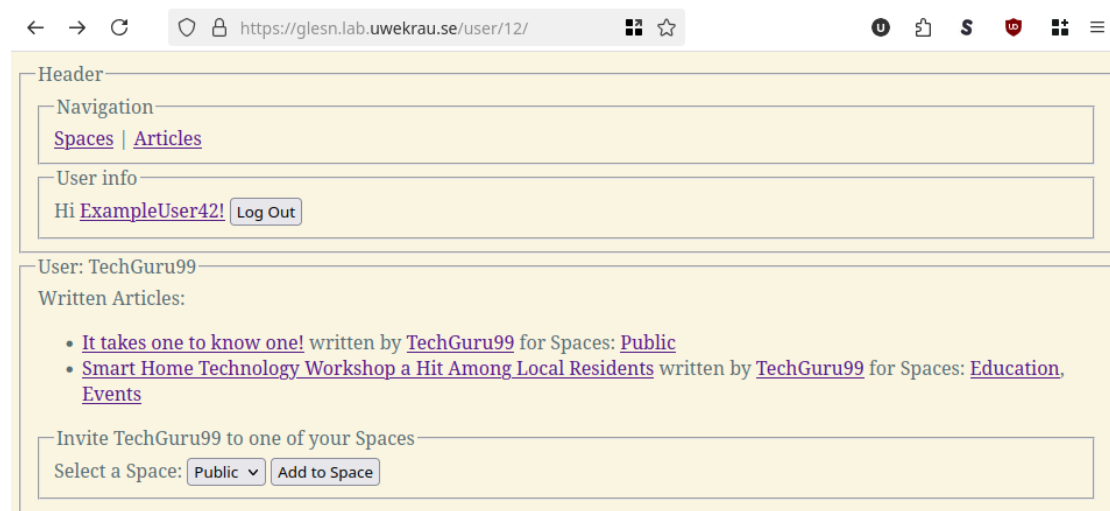


Abbildung 4: Screenshot der Website mit Auflistung der von einer Person verfassten Artikel. Die Person kann zu einem Space eingeladen werden.

3.4 Sicherheitslücken in der Anwendung

Nun besteht der eigentliche Zweck der Anwendung nicht in dem Austausch von Nachrichten, sondern darin, den zu lehrenden Inhalten einen mehr oder weniger realistischen Rahmen zu geben. Dem Szenario folgend hat „die Entwickler:in der Plattform“ mehrere Fehler gemacht, die zu ausnutzbaren Sicherheitslücken führen.

Programmierer:innen von Web-Anwendungen unterliegen häufig dem Irrtum, dass die Bedienung nur mit den von der Anwendung selbst bereitgestellten Mitteln möglich ist (CWE425), weshalb zwei Sicherheitslücken dieser Art hinterlegt werden.

3.4.1 Auslesen von Meta-Daten trotz fehlender Zugriffsberechtigung

Im Beispiel der Glesn-Anwendung werden sowohl in der Spaces-, als auch in der Artikel-Übersicht nur Links zu Ressourcen-Übersichten aufgelistet, auf die die Anwender:in Zugriff hat. Die vergebenen Berechtigungen (Zugehörigkeit zum Space) werden beim Aufbau der Bedienoberfläche also respektiert.

Über einen direkten Aufruf der URL ist jedoch ein gezielter Zugriff auf die Übersichtsseite einer ausgewählten Ressource möglich. Wird in der Glesn-Anwendung eine Artikel-Ansicht direkt aufgerufen, wird auch auf der Übersichtsseite noch einmal die Berechtigung überprüft. Der Inhalt des Artikels bleibt also trotzdem geschützt. Die Übersichts-Seite zeigt jedoch Meta-Daten über den Artikel an: Angezeigt werden die Spaces, denen dieser Eintrag zugewiesen ist und die Autor:in des Artikels. Da die Personen-Übersicht nur Artikel anzeigt, zu denen eine Anwender:in Zugriff hat, sich jedoch über den genannten Umweg trotzdem zumindest alle Artikel-Überschriften zu Autor:innen zuweisen lassen, ist dies als Informationsleck (CWE200) zu betrachten.

3.4.2 Rechteausweitung

Wie beschrieben bietet die Anwendung die Möglichkeit, andere Benutzer:innen zu Spaces einzuladen. Durch eine erste Berechtigungsprüfung sind in der grafischen Oberfläche in einer Auswahl-Liste für die Anwender:in nur diejenigen Spaces zum Einladen verfügbar, zu denen Zugriff besteht.

Eine Angreifer:in kann die Oberfläche jedoch umgehen und beim Web-Server direkt eine Einladung anfragen. Wieder wird die Annahme ausgenutzt, dass die vom Web-Server angebotenen Endpunkte ausschließlich wie von der Entwickler:in vorgesehen verwendet werden (CWE425). In der Beispiel-Anwendung wurde davon ausgegangen, dass der Endpunkt durch die erste Berechtigungsprüfung bereits geschützt ist. In diesem Szenario wird deshalb fehlendes Bewusstsein dahingehend unterstellt, dass vor dem erfolgreichen Aussprechen einer Einladung zu einem Space überprüft werden muss, ob die einladende Person überhaupt selbst über die notwendige Berechtigung verfügt.

Durch die fehlende Autorisationsprüfung werden mehrere Schutzziele der Anwendung gefährdet: Eine Angreifer:in kann eine privilegierte Funktion aufrufen, die Daten in ihrem

Sinne verändert. Auf diese Weise erhält sie Zugriff auf geschützte Informationen. Durch Verletzung der Integrität werden Zugriffskontrollen umgangen und die Vertraulichkeit der Anwendung verletzt (CWE862).

3.5 Ausnutzen der Sicherheitslücken

Zu dieser Art von Aufgaben gehört ein angedachter Lösungsweg, eine Art Ablaufplan, von dem zumindest erwartet wird, dass die Angreifer:innen ungefähr diesem Pfad folgen. Die Challenge lässt sich lösen, indem die innerhalb der Anwendung platzierten Schwachstellen ausgenutzt werden. Die Sicherheitslücken müssen miteinander kombiniert werden, um ausgehend von einer unauthentifizierten Anwender:in zum Ziel der Rechteaussweitung zu kommen.

Besondere Angriffswerkzeuge werden nicht vorausgesetzt. Mit dem in modernen Webbrowsern vorhandenem Werkzeug der „Netzwerkanalyse“, können die einzelnen Aufrufe auf technischer Ebene protokolliert, eingesehen und erneut versendet werden. Einige Web-Browser (z.B. Firefox) erlauben an dieser Stelle auch eine Manipulation der Anfrage vor dem Wiederversenden.

Als Erstes wird davon ausgegangen, dass Angreifer:innen ohne eigenen Nutzer-Zugriff die Anwendung auskundschaften. Hierbei sollte schon ohne Account auffallen, dass die öffentlich sichtbaren Spaces und Artikel anhand ihrer ID aufgerufen werden. Der sichtbare Space „Public“ hat die ID 2 und im Browser wird als Navigationspfad `/space/2/` angezeigt. Der darin abgelegte Artikel hat die Artikel-ID 2 und wird über `/article/2/` aufgerufen. Die Oberfläche zeigt unter anderem auch einen Link zur Übersichtsseite der Autor:in des Artikels an, die sich in diesem Beispiel über `user/12/` aufrufen lässt. Abbildung 5 zeigt einen Ausschnitt der Netzwerkanalyse.

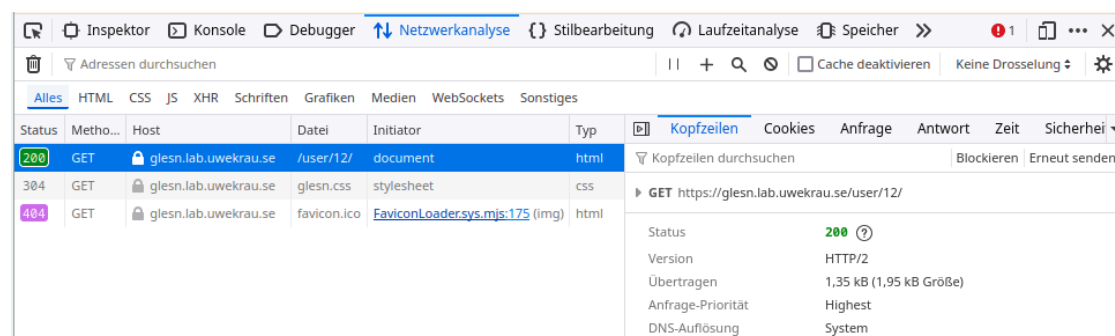


Abbildung 5: Screenshot der Netzwerkanalyse: Erfolgreicher GET Aufruf auf `/user/12/`.

Dass die ersten sichtbaren Einträge die ID 2 haben, soll als Hinweis verstanden werden, dass es unter den weiteren vorhandenen Spaces, Artikeln und User-Accounts auch jeweils Objekte mit der ID 1 gibt, die von besonderem Interesse sein könnten. Durch einen direkten Aufruf der Ressourcen, in Kombination mit der Anzeige der unzureichend geschützten Meta-Daten, können Informationen über weitere Spaces, Artikel und Autor:innen nach und nach ausgelesen (*enumeriert*) werden. So enthalten die Beispieldaten unter anderem ein Account mit dem Namen „admin“, ein Space dessen Name „!!! ADMIN ONLY !!!“ lautet und ein Artikel mit der Überschrift „Note FOR ADMINS ONLY !!!“. Diese sprechenden Namen sollen ein weiterer Hinweis auf das zu erreichende Ziel sein und den Teilnehmer:innen der ctf-Challenge somit Orientierung bieten.

Nun sollte die nächste zur Verfügung stehende Funktionalität der Website verwendet werden: Die Registration eines neuen Accounts. Mit diesem Account sind zwar keine zusätzlichen Spaces oder Artikel sichtbar, aber auf der Informationsseite zu anderen Autoren erscheint nun – wie zuvor in Abbildung 4 gezeigt – die Möglichkeit, die andere Person zu Spaces einzuladen. Mit Hilfe der Netzwerkanalyse kann dem Protokoll entnommen werden, dass eine Einladung zu einem Space auf technischer Ebene durch ein POST-Request angefordert wird. Dieser Anfrage wird als Argument die ID des Spaces über den Parameter `space_id` übergeben, wie in Abbildung 6 dargestellt.

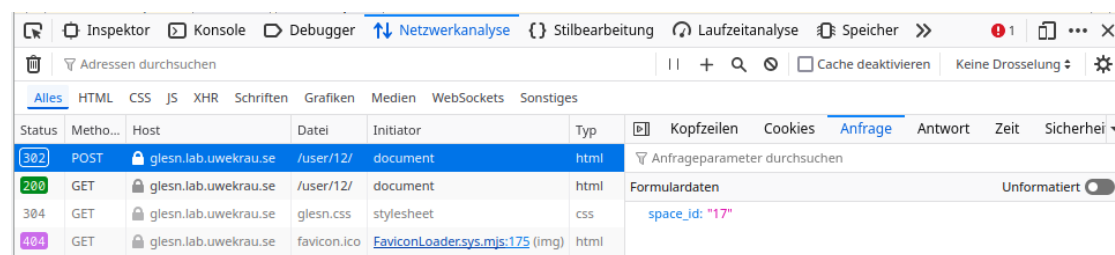


Abbildung 6: Screenshot der Netzwerkanalyse: Erfolgreicher POST Aufruf auf `/user/12/` mit Argument `space_id: "17"`.

Der angedachte Lösungsweg sieht vor, dass eine Angreifer:in nun versuche sollte, die Parameter der Anfrage zu manipulieren und zu probieren, verschiedenen Accounts zu verschiedenen Spaces einzuladen. Aufgrund der Sicherheitslücke der fehlenden Berechtigungsprüfung wird dies gelingen und die Angreifer:in kann sich selbst durch eine manipulierte POST-Anfrage zu dem Space mit der `space_id: "1"` einladen. Da sie nun Mitglied des Spaces ist, also ihre eigenen Zugriffsrechte erhöht hat, kann sie über die reguläre Benutzeroberfläche auch den Inhalt der eigentlich geschützten Notiz lesen.

3.6 Beta-Test

Während der Entwicklung dieser Challenge wurde ein Beta-Test durchgeführt, bei dem sich mehrere Mitglieder des lokalen CTF-Teams Cyclopropenylidene (C3H2) bereit erklärt haben, eine Vorabversion der Aufgabe zu lösen. Aus den gesammelten Rückmeldungen sind zwei Punkte besonders nennenswert:

In einem früheren Entwicklungsstand der Challenge wurden in die Datenbank der Beispieldaten zunächst die Artikel eingefügt, dann erst der zu findende Artikel mit den versteckten Informationen (siehe Abschnitt 3.5). Dies führte dazu, dass dieser Artikel eine unvorhersehbare ID wie 17 oder 19 hatte, wodurch wiederum einzelnen Beta-Tester:innen das Ziel nicht deutlich genug war. Einer der Beta-Tester hat angemerkt, dass die IDs als eine Art Hinweis verwendet werden könnten, was in einer späteren Version der Challenge umgesetzt wurde.

Außerdem wurde ein technisches Problem aufgedeckt: Ein Beta-Tester hat nicht nur einen Account angelegt und (wie angedacht) diesem Account Zugriff auf den versteckten Space gegeben, sondern mit Hilfe eines *http replay proxy* die Sicherheitslücke so ausgenutzt, dass alle Accounts Zugriff auf den versteckten Space hatten. Das hat dafür gesorgt, dass eine Beta-Testerin, die sich zuvor einen Account angelegt hat, aber die Sicherheitslücke noch nicht gefunden hat, plötzlich ohne eigenes Zutun Zugriff auf den versteckten Space hatte. Für diese Challenge ist es also notwendig, sie pro Person/Team als eigene isolierte Instanz bereitzustellen.

3.7 Durchführung eines Workshops

Während der sogenannten „[Workshop Week]“ an der Hochschule für Angewandte Wissenschaften (HAW) Hamburg im September 2024 wurden mit Hilfe diese Challenge in einer halbtägigen Veranstaltung ca. 12 Teilnehmer:innen in das Problemfeld der mangelhaften Zugriffskontrolle eingeführt. Aus Reaktionen während der Veranstaltung und in Gesprächen nach der Veranstaltung wurde ersichtlich, dass einzelne Teilnehmer:innen gelernt haben, wie leicht http-Anfragen wie GET und POST manipuliert werden können und wie wichtig daher eine serverseitige Überprüfung der Nutzereingaben ist. Zusätzlich dazu wurde ein deutliches Interesse an dieser Lernform zu IT-Sicherheitsthemen geäußert.

4 Fazit & Ausblick

4.1 Fazit

In dieser Arbeit wurde aufbauend auf der didaktischen Basis von Blooms Taxonomie eine Methode gesucht, mit dem Ziel, das Bewusstsein für Sicherheitsprobleme bei der Programmierung von IT-Systemen zu steigern. Während der ICAP-Ansatz eine möglichst starke Interaktion mit dem Lerngegenstand nahelegt, interagieren Anwender:innen vertieft mit Unterhaltungsmedien, weshalb eine Methode aus dem Unterhaltungsbereich entliehen wurde. Als Teil eines Serious Games wurde eine interaktive Lernumgebung entwickelt, anhand derer Teilnehmer:innen eines Workshops selbst erleben konnten, welche Auswirkungen Nachlässigkeit bei der Programmierung von IT-Systemen haben kann.

In dem fiktiven Szenario einer Programmierer:in, die die Zugriffskontrolle einer Web-Anwendung unzureichend abgesichert hat, konnten die Teilnehmer:innen in die Rolle einer Angreifer:in schlüpfen und mit frei zugänglichen Werkzeugen diese Sicherheitslücken selbst ausnutzen. Aus den Rückmeldungen der Teilnehmer:innen war Lernerfolg und gesteigertes Interesse für interaktives Lernen von IT-Sicherheitsthemen und capture the flag als Methode ersichtlich.

4.2 Ausblick

Aufbauend auf und ergänzend zu diesem Beispiel sollten weitere Empfehlungen für Verbesserungen und Erkenntnisse zur Gestaltung und Entwicklung effektiverer serious games (WANICK u. a., 2023) betrachtet werden, um das Sicherheitsbewusstsein von Programmierer:innen zu steigern. Als konkretes Beispiel könnten Evaluations-Methoden zur User Experience aus der allgemeinen Spiele-Entwicklung oder dem Übergeordneten Feld der serious games auch auf die entwickelte Challenge adaptiert werden (KIRGINAS, 2023).

Die entwickelte Challenge sollte daraufhin überprüft werden, ob sie formulierten Anforderungen an Lernumgebungen für serious games (ESPINHA GASIBA u. a., 2019) entspricht. Bei einer weiteren zu entwickelnden Challenge sollten die dort formulierten Anforderungen von Anfang an integriert werden.

Stehen mehrere Challenges zur Verfügung, können diese für ein „Capture The Flag“-Event verwendet werden.

Bibliografie

- ANDERSON, L. W. ; KRATHWOHL, D. R. (Hrsg.): *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Complete ed.. New York : Longman, 2001 — ISBN 978-0-321-08405-7 978-0-8013-1903-7
- COOPER, K. M. L. ; BUCCHIARONE, A. (Hrsg.): *Software Engineering for Games in Serious Contexts*. Cham : Springer Nature Switzerland, 2023 — ISBN 978-3-031-33337-8 978-3-031-33338-5
- CHI, MICHELENE T. H. ; WYLIE, RUTH: The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. In: *Educational Psychologist* Bd. 49 (2014), Nr. 4, S. 219–243
- COSTA, GABRIELE ; RIBAUDO, MARINA: Designing a Serious Game for Cybersecurity Education, COOPER, K. M. L. ; BUCCHIARONE, A. (Hrsg.). *Software Engineering for Games in Serious Contexts*.
- EAGLE, CHRIS ; CLARK, JOHN L.: Capture-the-Flag: Learning Computer Security Under Fire, Monterey, California: Naval Postgraduate School (2004)
- ECKSTEIN, JUTTA ; BERGIN, JOSEPH ; SHARP, HELEN: Patterns for Active Learning.. In: *Proceedings of PLOP*. Bd. 2002, 2002
- ESPINHA GASIBA, TIAGO ; BECKERS, KRISTIAN ; SUPPAN, SANTIAGO ; REZABEK, FILIP: On the Requirements for Serious Games Geared Towards Software Developers in the Industry.. In: *2019 IEEE 27th International Requirements Engineering Conference (RE)*. Jeju Island, Korea (South) : IEEE, 2019 — ISBN 978-1-7281-3912-8, S. 286–296
- IBM DEUTSCHLAND: *Cost of a Data Breach 2024*. URL <https://www.ibm.com/reports/data-breach>. - abgerufen am 2025-04-14
- KIRGINAS, SOTIRIS: User Experience Evaluation Methods for Games in Serious Contexts, COOPER, K. M. L. ; BUCCHIARONE, A. (Hrsg.). *Software Engineering for Games in Serious Contexts*.
- LEUNE, KEES ; PETRILLI, SALVATORE J.: Using Capture-the-Flag to Enhance the Effectiveness of Cybersecurity Education.. In: *Proceedings of the 18th Annual Conference on Information Technology Education*. Rochester New York USA : ACM, 2017 — ISBN 978-1-4503-5100-3, S. 47–52

- MCDANIEL, LUCAS ; TALVI, ERIK ; HAY, BRIAN: Capture the Flag as Cyber Security Introduction.. In: *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 2016, S. 5479–5486
- MITRE: *Common Weakness Enumeration - CWE Version 4.16*. URL https://cwe.mitre.org/data/published/cwe_v4.16.pdf. - abgerufen am 2025-02-25
- OWASP: *A01 Broken Access Control - OWASP Top 10:2021*. URL https://owasp.org/Top10/2021/A01_2021-Broken_Access_Control/. - abgerufen am 2025-02-25
- OWASP: *A01 Broken Access Control - OWASP Top 10:2025*. URL https://owasp.org/Top10/2025/A01_2025-Broken_Access_Control/. - abgerufen am 2025-12-15
- REDDIT.COM: *Faq* - *Reddit.Com*. URL https://web.archive.org/web/20201127015655/http://www.reddit.com/wiki/faq#wiki_what_does_the_name_.22reddit.22_mean.3F. - abgerufen am 2025-02-26
- SIAMI NAMIN, AKBAR ; AGUIRRE-MUÑOZ, ZENaida ; JONES, KEITH: Teaching Cyber Security through Competition An Experience Report about a Participatory Training Workshop.. In: *7th Annual International Conference on Computer Science Education: Innovation & Technology (CSEIT 2016)* : Global Science & Technology Forum (GSTF), 2016
- TSEPILIS, VASILEIOS: *A Beginner's Guide to CTFs : Learning Cybersecurity Through Capture The Flag Challenges*. URL <http://www.theseus.fi/handle/10024/898463>. - abgerufen am 2025-12-11. — Accepted: 2025-10-08T10:05:15Z
- WANICK, VANISSA ; STALLWOOD, JAMES ; XAVIER, GUILHERME: Future Directions in Games for Serious Contexts: A Conversation About Transferability, COOPER, K. M. L. ; BUCCHIARONE, A. (Hrsg.). *Software Engineering for Games in Serious Contexts*.